

# 基于相似函数与相似网络的可逆网络化简

徐明强<sup>1,2</sup> 管致锦<sup>2</sup> 何金凤<sup>3</sup> 鹿玉<sup>2</sup>

(南通纺织职业技术学院素质部 南通 226007)<sup>1</sup> (南通大学计算机科学与技术学院 南通 226019)<sup>2</sup>  
(南通大学电子信息学院 南通 226019)<sup>3</sup>

**摘要** 提出了可逆函数的相似函数及可逆网络的相似网络,在此基础上构建了可逆网络化简方法。由可逆函数求出其所有的相似函数,对每个相似函数利用可逆逻辑综合算法生成可逆网络,再将其转换成对应的相似网络,并从中选取最优。该网络化简算法实现了生成三变量全部可逆函数和多变量可逆函数的可逆网络,与相关文献及 Benchmark 中的例题相比,构造可逆网络的门数较少,具有一定的优势。

**关键词** 可逆逻辑,可逆网络,相似函数,相似网络

**中图分类号** TP302.2 **文献标识码** A

## Reversible Network Simplification with Similar Function and Similar Network

XU Ming-qiang<sup>1,2</sup> GUAN Zhi-jin<sup>2</sup> HE Jin-feng<sup>3</sup> LU Yu<sup>2</sup>

(Basic Department, College of Nantong Textile Vocational Technology, Nantong 226007, China)<sup>1</sup>

(College of Computer Science and Technology, Nantong University, Nantong 226019, China)<sup>2</sup>

(College of Electronics and Information, Nantong University, Nantong 226019, China)<sup>3</sup>

**Abstract** This paper presented the similar function of the reversible function and the similar network of the reversible network, and on the basis of that, the simplification method of reversible network was constructed. Giving a reversible function, all its similar functions can be searched. To every similar function, the reversible network which is converted from its corresponding similar network, can be constructed by reversible logic synthesis algorithm, and the optimum can be chosen. The network simplification algorithm realizes the reversible network of all 3-variable functions and some multi-variable functions. Compared with the pertinent literature and the examples of Benchmark, it has some advantages while constructing the reversible network with less gate count.

**Keywords** Reversible logic, Reversible network, Similar function, Similar network

## 1 引言

可逆计算避免了计算过程中比特位的丢失,理论上能将计算中的逻辑能耗降低为零<sup>[1]</sup>,在低功耗电路设计、量子计算、光计算等领域有着广泛的应用。为了实现可逆计算,如何使用给定的可逆逻辑门级联生成可逆网络,成为可逆逻辑综合研究的重点和难点。为此,国内外相关研究学者相继提出了许多高效的可逆逻辑综合算法:管致锦等<sup>[2]</sup>提出了可逆网络的表示,在此基础上给出相应的网络级联算法;王冬等<sup>[3]</sup>提出了基于矩阵初等变换的可逆电路综合算法;李志强等<sup>[4]</sup>通过构造置换最短编码,对可逆电路采用拓扑变换等方法生成最优的 4 量子可逆电路;Pang Yu 等<sup>[5]</sup>提出基于 PDD 判决图的可逆电路综合算法;Shende 等<sup>[6]</sup>提出了一种基于三变量的最优综合算法;Gupta 等<sup>[7]</sup>提出了基于 PPRM 扩展的方法;Maslov 等<sup>[8]</sup>提出了利用模版技术综合的方法;Saedi 等<sup>[9]</sup>提出了基于真值表的行调换方法;Zheng Yexin 等<sup>[10]</sup>提出了利

用 POT 转换技术的方法。利用这些方法可以生成网络代价较小甚至最优的可逆网络,对可逆网络的生成与优化具有重要意义。

文献[4]提出了对量子电路的量子线路的拓扑变化,阐述了通过拓扑变换后特定的不同量子电路之间的关系。由此,与之对应的不同的可逆函数,本文从中尝试建立函数之间的关系并搜索出其相似函数。对可逆函数的综合,可以转化成对该函数及其相似函数的综合,级联生成若干可逆网络,从中选择网络代价最小的可逆网络并对其转化成与函数对应的相似网络,从而化简可逆网络。

## 2 基本概念

### 2.1 可逆函数

**定义 1**  $n$  输入  $n$  输出的函数  $F(x_1, x_2, \dots, x_n)$  是可逆函数,则  $n$  位输入向量  $(x_1, x_2, \dots, x_n)$  一一映射到  $n$  位输出向量  $(y_1, y_2, \dots, y_n)$ 。

本文受国家自然科学基金(60873069),南通纺织职业技术学院科研项目(FYKY/2013/7),南通大学研究生科技创新计划项目(YKCl3006)资助。  
徐明强(1982—),男,硕士,讲师,主要研究方向为可逆计算、逻辑综合及多值可逆逻辑等;管致锦(1962—),男,博士,教授,主要研究方向为可逆计算、逻辑综合和信息安全等,E-mail:guan\_zj@ntu.edu.cn(通信作者);何金凤(1989—),女,硕士生,主要研究方向为可逆逻辑综合和可逆网络控制与检测等;鹿玉(1991—),女,硕士生,主要研究方向为量子可逆电路逻辑综合、线性最近邻可逆逻辑综合。

将  $n$  位可逆函数记为  $F_n$ , 通常可以由整数序列  $[0, 1, 2, \dots, 2^n - 1]$  表示, 完成由  $n$  个输入变量到  $n$  个输出变量的一一映射。

## 2.2 可逆逻辑门

**定义 2** 一个  $n$  输入  $n$  输出的逻辑门为可逆逻辑门, 则该逻辑门实现可逆函数的一个映射。

可逆逻辑门是组成可逆网络的基本单位。将若干相同或不同的可逆逻辑门按照无反馈、无扇出的规则级联生成可逆网络, 可以实现任意可逆函数。

**定义 3** NOT( $x$ ) 门是 1 输入 1 输出可逆逻辑门, 完成  $(x) \rightarrow (x \oplus 1)$  的映射 ( $\oplus$  为异或运算)。

**定义 4** CNOT( $x_1, x_2$ ) 门是 2 输入 2 输出可逆逻辑门, 完成  $(x_1, x_2) \rightarrow (x_1, x_1 \oplus x_2)$  的映射。

**定义 5** TOFFOLI( $x_1, x_2, x_3$ ) 门是 3 输入 3 输出可逆逻辑门, 完成  $(x_1, x_2, x_3) \rightarrow (x_1, x_2, x_1 x_2 \oplus x_3)$  的映射。

**定义 6**  $n$ TOFFOLI( $x_1, x_2, \dots, x_{n-1}, x_n$ ) 门是  $n$  输入  $n$  输出可逆逻辑门, 完成  $(x_1, x_2, \dots, x_{n-1}, x_n) \rightarrow (x_1, x_2, \dots, x_{n-1}, x_1 x_2 \dots x_{n-1} \oplus x_n)$  的映射 ( $n > 3$ )。

**定义 7** 可逆逻辑门集是指用于构建可逆网络级联的一组可逆逻辑门。

在可逆逻辑综合过程中需要选取一组可逆逻辑门对可逆函数进行综合, 生成可逆电路。目前, 常用可逆逻辑门集主要有 NCT 门集, NCTSF 门集, GF 门集, GT 门集<sup>[1]</sup> 和 PNC 门集<sup>[3]</sup> 等。

## 3 可逆网络化简

### 3.1 相似函数

**定义 8** 变量置换  $\sigma: \sigma(x_1, x_2, \dots, x_n) = (x_{j_1}, x_{j_2}, \dots, x_{j_n})$ ,  $j_i \in [1, n]$ , 当  $k \neq m$  时,  $j_k \neq j_m$ 。

如  $\sigma(x_1, x_2, x_3) = (x_2, x_3, x_1)$  就是一个三变量置换。在三变量可逆函数中, 更替变量的前后位置顺序, 其变量置换共有 6 种, 分别为:  $(x_1, x_2, x_3)$ ,  $(x_1, x_3, x_2)$ ,  $(x_2, x_1, x_3)$ ,  $(x_2, x_3, x_1)$ ,  $(x_3, x_2, x_1)$ ,  $(x_3, x_1, x_2)$ 。

**定理 1** 对于  $n$  变量可逆函数, 其变量置换  $\sigma$  的个数为  $n!$  个。

证明: 对于  $n$  个不同变量  $(x_1, x_2, \dots, x_n)$  排列的个数是排列组合中的排列问题, 共有  $P_n$  种, 所以其变量置换的个数为  $P_n = n!$ 。

**定义 9** 相似函数  $F_s$ : 对于  $n$  变量可逆函数  $F(x_1, x_2, \dots, x_n) = (y_1, y_2, \dots, y_n)$ ,  $\sigma$  是  $n$  变量置换, 则其相似函数  $F_s(\sigma(x_1, x_2, \dots, x_n)) = \sigma(y_1, y_2, \dots, y_n)$ 。

相似函数的个数是由变量置换  $\sigma$  的个数决定的。由定理 1、定义 8 和定义 9 可知,  $n$  变量可逆函数的变量置换  $\sigma$  的个数为  $n!$ , 则其相似函数最多也是  $n!$ 。这是因为存在一些可逆函数, 的全部或部分相似函数相等的情况, 即  $F_{s_k}(\sigma_k(x_1, x_2, \dots, x_n)) = F_{s_m}(\sigma_m(x_1, x_2, \dots, x_n))$ , 如三变量可逆函数  $[0, 1, 2, 3, 4, 5, 7, 6]$ ,  $\sigma_0 = (x_1, x_2, x_3)$ ,  $\sigma_1 = (x_1, x_3, x_2)$ ,  $F_{s_0}(\sigma_0) = F_{s_1}(\sigma_1) = [0, 1, 2, 3, 4, 5, 7, 6]$ ;  $\sigma_2 = (x_2, x_1, x_3)$ ,  $\sigma_3 = (x_2, x_3, x_1)$ ,  $F_{s_2}(\sigma_2) = F_{s_3}(\sigma_3) = [0, 1, 2, 3, 4, 7, 6, 5]$ ;  $\sigma_4 = (x_3, x_1, x_2)$ ,  $\sigma_5 = (x_3, x_2, x_1)$ ,  $F_{s_4}(\sigma_4) = F_{s_5}(\sigma_5) = [0, 1, 2, 7, 4, 5, 6, 3]$ 。对于可逆函数  $[0, 1, 2, 3, 4, 5, 7, 6]$ , 其相似函数的个数只

有 3; 对于可逆函数  $[7, 6, 5, 4, 3, 2, 1, 0]$ , 其相似函数的个数仅为 1, 相似函数为其本身, 即可逆函数  $[7, 6, 5, 4, 3, 2, 1, 0]$ 。

**推论 1** 对于  $n$  变量可逆函数, 其相似函数的个数最多为  $n!$ 。

$n$  位可逆函数的个数达到  $2^n!$ 。当可逆函数中变量规模线性增大时, 其相似函数个数成阶乘增加。可逆函数的变量规模  $n=3$ , 其相似函数个数最多为 6; 当  $n=8$  时, 可逆函数的相似函数最多可以达到  $8! = 40320$  个。

根据可逆函数, 如何求得其相似函数是本文关键。其基本思想: 根据需要置换的变量, 首先找出置换行, 相互交换这两行的输出  $Y = (y_n, y_{n-1}, \dots, y_1)$ , 因为置换变量的先后顺序发生了变化, 所以还需要置换这两个变量的顺序, 下面给出相似函数的求解算法:

### 算法 1 相似函数 SimilarFun

输入: 可逆函数  $F$ , 置换的变量  $y_j, y_k$  序号  $j$  和  $k$

输出: 相似函数  $F_s$

第一步 初始化:  $i=0, \max = \max(j, k), \min = \min(j, k)$ ; //  $\max(j, k)$  返回  $j$  和  $k$  的最大值,  $\min(j, k)$  返回  $j$  和  $k$  的最小值

第二步 if  $(x_{i, \min} = 1 \& \& x_{i, \min} = x_{i, \max})$

SWAP( $Y_i, Y_{i+2^{\max-1}-2^{\min-1}}$ ); // 交换  $i$  行和  $i+2^{\max-1}-2^{\min-1}$  行的输出  $Y$

Swap( $y_{i, k}, y_{i, j}$ ); // 交换  $i$  行  $y_k$  和  $y_j$  的值

第三步  $i++$ ;

if  $(i < 2^n)$  返回第二步;

第四步: 输出相似函数  $F_s$ 。

如可逆函数  $F = [7, 0, 1, 2, 3, 4, 5, 6]$ , 表 1 中, 分别使用上述算法 SimilarFun( $F, 2, 3$ )、SimilarFun( $F, 1, 2$ )、SimilarFun( $F_{s_2}, 1, 3$ )、SimilarFun( $F, 1, 3$ ) 和 SimilarFun( $F_{s_4}, 1, 2$ ) 求解, 可得其相似函数  $F_{s_1} = [7, 0, 5, 2, 1, 4, 3, 6]$ 、 $F_{s_2} = [7, 2, 0, 1, 3, 6, 4, 5]$ 、 $F_{s_3} = [7, 6, 0, 1, 2, 3, 4, 5]$ 、 $F_{s_4} = [7, 4, 5, 6, 0, 1, 2, 3]$  和  $F_{s_5} = [7, 6, 4, 5, 0, 1, 2, 3]$ 。

表 1 相似函数

可逆函数 $F$	相似函数				
	$F_{s_1}$	$F_{s_2}$	$F_{s_3}$	$F_{s_4}$	$F_{s_5}$
$[7, 0, 1, 2, 3, 4, 5, 6]$	$[7, 0, 5, 2, 1, 4, 3, 6]$	$[7, 2, 0, 1, 3, 6, 4, 5]$	$[7, 6, 0, 1, 2, 3, 4, 5]$	$[7, 4, 5, 6, 0, 1, 2, 3]$	$[7, 6, 4, 5, 0, 1, 2, 3]$

### 3.2 相似网络

**定义 10**(相似网络) 相似网络  $G_s$  置换可逆网络  $G$  中网络线路的位置, 同时保持对应可逆门与网络的交叉点(包括控制点和受控点), 而不改变网络中可逆门的先后顺序。

可逆网络  $G$  中的线路经过线路置换(变量置换)得到它的相似网络  $G_s$ , 记为  $G \rightarrow G_s$ 。

**定理 2** 设可逆网络  $G$  是  $n$  变量可逆函数  $F$  的一个网络实现, 该函数的相似函数为  $F_s$ , 可逆网络  $G$  的相似网络为  $G_s$ , 相似函数的个数为  $C_{F_s}$ , 相似网络的个数为  $C_G$ , 则  $C_{F_s} = C_G$ 。

证明: 可逆函数  $F$  由变量置换  $\sigma$  后, 得到其相似函数  $F_s$ , 将其可逆网络  $G$  中的线路(变量)置换  $\sigma$  后, 得到与相似函数  $F_s$  相对应的相似网络  $G_s$ , 所以相似函数和其相似网络是一一对应的, 即  $C_{F_s} = C_G$ 。

**推论 2** 对于  $n$  变量可逆网络  $G$ , 其相似网络  $G_s$  的个数最多为  $n!$ 。

一般情况下, 对于  $n$  位可逆网络, 改变变量在可逆网络线

路中的位置,其相似网络的个数可以达到  $n!$ 。如可逆函数  $[7,0,1,2,3,4,5,6]$  的一个可逆网络实现如图 1(a)所示,该网络的其余 5 个可逆网络如图 1(b)~图 1(f)所示。需要注意的是,与相似函数类似,并不是所有的  $n$  位可逆网络都有  $n!$  种,如单个 NOT 门和单个 TOFFOLI 门所组成的 3 位可逆网络,它们的相似网络个数只有  $C_3^3=3$  种,由单个 NOT 门和单个 TOFFOLI 门所组成的 4 位可逆网络,它们的相似网络个数分别是  $C_4^1=4$  和  $C_4^1C_3^3=12$ 。

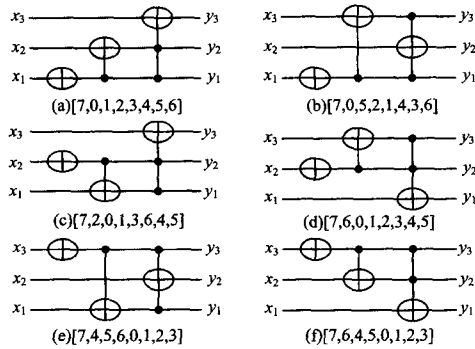


图 1 相似网络

### 3.3 可逆网络化简

可逆网络的化简主要由分为以下步骤完成。首先由上述给出的相似函数求解算法求出可逆函数的相似函数;接着选取可逆逻辑综合算法对每个相似函数进行综合,求得相应的可逆网络  $G_i$ ;最后从中选出门数最小的可逆级联网络  $G$ ,完成可逆网络向相似网络的线路置换  $G \rightarrow G_s, G_s$  为化简后的可逆网络。下面给出可逆网络的化简算法:

#### 算法 2 网络化简 Network Simplification

输入:可逆函数  $F$

输出:可逆网络  $G$

第一步 由相似函数求解算法分别求出可逆函数  $F$  的相似函数  $F_{s1}, \dots, F_{sk}$ ;

第二步 使用可逆逻辑综合算法分别求出相似函数的可逆网络  $G_{s1}, \dots, G_{sk}$ ;

第三步 分别计算可逆网络  $G_{s1}, \dots, G_{sk}$  的门数  $C_{s1}, \dots, C_{sk}$ , 选出门数  $C_{si}$  最小的可逆网络  $G_{si}$ ;

第四步 完成可逆网络向其相似网络的线路置换  $G_{si} \rightarrow G$ ;

第五步 输出可逆网络  $G$ 。

上述算法中第二步对可逆逻辑综合算法的选取并没有明确,即该化简算法可以适用不同的可逆逻辑综合算法对可逆网络进行化简优化,搜索网络代价更小的可逆网络。可逆逻辑综合算法的性能也间接影响着网络化简的结果,文献[13]提出的关联选择算法的复杂度较低,算法所生成的可逆网络代价较小,因此本文选取了文献[13]提出的关联选择算法作为该网络化简中第二步的可逆逻辑综合算法。

有两点需要注意,1)在对相似函数使用综合算法之前,必须把函数的输入输出与相似函数的输入输出先对应上,即确定变量置换关系,需要整个可逆网络的输入输出线路映射到其对应的相似网络的输入输出线路。如对可逆函数  $[7,0,1,2,3,4,5,6]$  进行综合时,假设最终得到的如图 1(b)所示的可逆网络是由相似函数  $[7,4,5,6,0,1,2,3]$  综合得到的,则需要将它们的输入输出线通过线路置换调整成如图 1(a)所示的可逆网络。2)一般对于  $n$  变量函数,其每个可逆函数的相似函数最多可以达到  $n!$  个,使得本网络化简的时间复杂度是文

献[13]的  $n!$  倍,这使我们很难承受。本文的做法是当  $n \leq 3$  时,找出所有的相似函数;当  $n \geq 4$  时,找出  $n$  个典型的相似函数,即循环置换可逆网络中变量的先后顺序:  $\sigma_i(x_1, x_2, \dots, x_n) = (x_{(1+i)\%n}, x_{(2+i)\%n}, \dots, x_{(n+i)\%n})$ , 这类变量置换共有  $n$  种,这样做可以充分改变相似函数的变量在网络中的线路位置。

使用本文的网络化简算法,可以搜索出比原可逆函数综合使用的可逆门更少的可逆网络。如函数  $[3,6,2,5,7,1,0,4]$  在文献[13]中实现的门序列为  $T(x_3, x_2, x_1), T(x_3, x_1, x_2), C(x_2, x_1), C(x_2, x_3), T(x_3, x_1, x_2), C(x_1, x_3), N(x_1), N(x_2)$ , 总共需要 8 个门。而在本文的网络化简算法实现的门序列为:  $C(x_2, x_3), T(x_1, x_2, x_3), T(x_3, x_2, x_1), T(x_3, x_1, x_2), C(x_1, x_3), N(x_2), C(x_2, x_1)$ , 共需 7 个门。

## 4 实验对比及分析

本文采用 C# 实现了网络化简算法,对综合网络的规模进行提升,可以对 10 变量以上的可逆网络进行综合。

### 4.1 三变量全部函数

对三变量全部函数进行综合,与文献[6,10,12,13]进行了比较,如表 2 所列。表 2 中,门数一列是综合全部函数所使用的门的个数,在门集一行中,为便于比较将不同文献所使用的门集进行归类,文献[6]是三变量综合的最佳算法,使用 NCT 门集,采用穷举法得到可逆网络,平均门数是 5.866,是目前综合三变量全部函数的最优结果,本文使用 NCT 门集的结果是 6.253,已经十分接近最佳算法。对比文献[13]中使用 NCT 门集的平均门数 6.806,本算法平均门数减少较为明显。与文献[10,12,13]中使用 PNC 门集综合的平均门数对比,本文使用 PNC 门集后平均门数降到 4.938,实验效果更好。

表 2 三变量全部函数

门数	文献	文献	文献	文献[13]		本文	
	[6]	[10]	[12]	NCT	PNC	NCT	PNC
0	1	1	1	1	1	1	1
1	12	12	27	12	27	12	27
2	102	90	282	102	327	102	339
3	625	476	1499	510	1882	589	2355
4	2780	1833	4471	1911	5618	2444	9383
5	8921	4996	8112	5058	10225	2960	16643
6	17049	9126	9731	9134	11454	12585	10422
7	10253	10630	8095	10610	7452	11829	1132
8	577	7820	4981	7911	2789	5042	18
9	0	3788	2241	3653	493	650	0
10	0	1265	699	1137	52	106	0
11	0	258	158	262	0	0	0
12	0	25	21	19	0	0	0
13	0	0	2	0	0	0	0
平均	5.866	6.834	6.141	6.806	5.657	6.253	4.938

### 4.2 3-4 位可逆函数

将具有代表性的一些 3 位和 4 位可逆函数综合生成可逆网络,并与文献[7,9,13]比较,结果如表 3 所列。表 3 中,所有算法都使用 GT 门集实现,可逆网络实现这列中列出了本文相应函数的可逆网络实现。表中 T3 表示 Toffoli 门, T4 表示 4-Toffoli 门。文献[7]的平均门数为 5.77,文献[9]的平均门数为 7.46,文献[13]的平均门数为 6.31,本文算法对这些可逆函数进行综合的平均门数为 5.62,平均门数比上述文献

的算法要低,本文的网络化简可以有效减少可逆逻辑综合中使用的门数。

表3 3-4位可逆网络

函数	文献[7]	文献[9]	文献[13]	本文算法
[1,0,3,2,5,7,4,6]	4	6	4	4
[7,0,1,2,3,4,5,6]	3	3	3	3
[0,1,2,3,4,6,5,7]	3	3	3	3
[0,1,2,4,3,5,6,7]	5	7	5	5
[0,1,2,3,4,5,6,8,7,9,10,11,12,13,14,15]	7	15	7	7
[1,2,3,4,5,6,7,0]	3	3	3	3
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0]	4	4	4	4
[0,7,6,9,4,11,10,13,8,15,14,1,12,3,2,5]	4	3	5	4
[3,6,2,5,7,1,0,4]	7	8	8	7
[1,2,7,5,6,3,0,4]	6	8	6	6
[4,3,0,2,7,5,6,1]	7	8	7	6
[7,5,2,4,6,1,0,3]	7	6	8	8
[6,2,14,13,3,11,10,7,0,5,8,1,15,12,4,9]	15	23	19	13
平均	5.77	7.46	6.31	5.62

### 4.3 4-8位随机函数比较

为了验证本文在多变数( $n>3$ )函数综合时网络的化简程度,将本文使用的相似函数化简后的综合算法与文献[13]对比,通过生成4-8位的随机可逆函数,分别使用GT门集和PNC门集进行网络综合,其结果如表4所列。表4中,随机选取了50000个4位可逆函数,10000个5位可逆函数,2000个6位可逆函数,500个7位可逆函数,100个8位可逆函数。可以看出,综合可逆网络所使用的门数随着可逆网络的扩大而增加,一般在同等情况下,用PNC门集去进行综合,所使用的门数要比GT门集使用的可逆门数少,经过本文对可逆网络化简后,所使用的门数比文献[13]有明显减少。可以看出当网络规模增加时,网络优化效果有所降低,这是因为随着网络规模增加,相应的综合生成可逆网络的时间呈指数增长,难以对函数的所有相似函数进行综合,考虑到时间因素只能从中挑选部分相似函数,随着网络规模 $n$ 的增加,挑选的相似函数仅占全部的 $1/(n-1)!$ ,所占比例急剧下降,在增加有限的时间开销的同时,有效减少了综合所需要的门数。

表4 4-8位随机函数对比

n位网络	文献13		本文算法		随机函数	减少比例	
	GT	PNC	GT	PNC		GT	PNC
8	608.3	550.1	585.3	527.5	100	96.2%	95.9%
7	266.5	236.5	251.5	222.1	500	94.4%	93.9%
6	113.8	97.9	104.4	88.9	2000	91.7%	90.8%
5	47.08	38.98	41.18	33.59	10000	87.5%	86.2%
4	18.52	14.82	15.43	12.13	50000	83.3%	81.8%

### 4.4 Benchmark 例题函数比较

Benchmark 例题<sup>[11]</sup>是目前国际上衡量可逆逻辑综合中网络代价的权威数据(网络代价一般用门数或量子代价计算)。从例题中选取一些函数,在GT门集下使用本文网络化简算法,生成可逆网络,对比如表5所列。在表5中通过对比例题中Maslov提出的算法<sup>[8]</sup>所得到的数据可以看出,本文提出的网络化简在网络规模 $n<8$ 时,网络代价无论是在门的个数还是在量子代价方面都相对落后,但是随着网络规模的增加,本文算法的网络代价无论在门数还是在量子代价都相对较小。

表5 Benchmark 例题函数对比

函数	Maslov		本文算法	
	门数	量子代价	门数	量子代价
Ham3	5	7	5	7
Ham7	21	69	29	119
Hwb6	42	150	97	1355
Hwb7	236	3984	240	4747
Hwb8	614	12745	579	14701
Hwb9	1541	43089	1315	38111
Hwb10	3595	136164	2910	102584
Hwb11	8214	336369	6414	235843

**结束语** 本文在文献[13]的基础上,提出并有效利用了相似函数和相似网络的特点,使得在综合可逆函数时能够化简综合网络,将该网络化简算法与其他文献算法对比,该算法在网络的门数代价方面有明显减少。下一步的工作目标为:优化该算法,使综合的网络规模进一步提高,降低网络的门数和量子代价并将其应用于多值可逆逻辑综合中。

### 参考文献

- [1] Bennett C H. Logical reversibility of computation[J]. IBM Journal of Research and Development, 1973, 17(6): 525-532
- [2] 管致锦,秦小麟,陶涛,等. 可逆逻辑门网络的表示与级联[J]. 电子学报, 2010, 38(10): 2370-2376
- [3] 王冬,陈汉武,安博,等. 基于矩阵初等变换的四量子比特可逆电路综合算法[J]. 电子学报, 2010, 38(11): 2561-2565
- [4] 李志强,陈汉武,徐宝文,等. 四量子可逆逻辑电路快速综合算法[J]. 电子学报, 2008, 36(11): 2081-2089
- [5] Pang Yu, Wang Shao-quan, He Zhi-long, et al. Positive Davio-based synthesis algorithm for reversible logic[C]//Proceedings of International Conference on Computer Design. New Jersey: Institute of Electrical and Electronics Engineers Inc., 2011: 212-218
- [6] Shende V V, Prasad A K, Markov I L, et al. Synthesis of reversible logic circuits [J]. Computer-Aided Design of Integrated Circuits and Systems, 2003, 22(6): 710-722
- [7] Gupta P, Agrawal A, Jha N K. An algorithm for synthesis of reversible logic circuits [J]. Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(11): 2317-2330
- [8] Maslov D, Dueck G W, Miller D M. Toffoli network synthesis with templates [J]. Computer Aided Design of Integrated Circuits and Systems, 2005, 24(6): 807-817
- [9] Saeedi M, Sedighi M, Zamani M S. A novel synthesis algorithm for reversible circuits[C]//Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society Press, 2007: 65-68
- [10] Zheng Y, Huang C. A novel Toffoli network synthesis algorithm for reversible logic[C]//Proceedings of the 2009 Asia and South Pacific Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2009: 739-744
- [11] Maslov D. Reversible logic synthesis benchmarks webpage [OL]. [2013-09]. <http://www.cs.uvic.ca/~dmaslov/>
- [12] Wan S, Chen H, Cao R. A novel transformation-based algorithm for reversible logic synthesis [C]//Proceedings of the 4th International Symposium on Advances in Computation and Intelligence. Heidelberg: Springer-Verlag Berlin 2009: 70-81
- [13] 徐明强,管致锦,倪丽惠. 基于关联选择的可逆逻辑综合算法[J]. 计算机辅助设计与图形学学报, 2012, 24(9): 1218-1225